

In the Specification

Page, 9, amend Paragraph 0042 as follows;

Symbol 9 represents a data structure or record type object. Again, the concept of a structure or record is well known. It is a standard feature of the C++ programming language, for instance. Generally, a data structure is a definition of a structure of data, e.g., field: name (string), field: age (integer), field: address (string). For instance, referring to Figure 2, each flight in frame ~~209~~ 290 (i.e., line ~~209a~~ 290a in frame ~~209~~ 290 is a populated record (or data structure). A data structure object is a static object. It can be assigned to another object, defined within another object or neither.

Page 11, amend Paragraph 0051 as follows:

As can be seen in the Figures, there are a number of buttons that the user may click upon in both windows 201a and 201b in order to cause something to occur. Referring first to the main window 201a (Figure 3), for instance, it includes the following buttons; NEW button 202, OPEN button 204, FLIGHTS button 206, CITIES button 208, PHONES button 210, PROGRAMS button 212, NEWS button 214, CALENDAR button 216, and HELP button 218. The window also includes a menu bar 221 including typical menus, e.g., FILE menu ~~222~~ 223, EDIT menu 225, TRAVEL menu 227, TRIP menu 229, OPTIONS menu 231, WINDOW menu 233, and HELP menu 235.

Page 14, amend Paragraph 0059 as follows:

Referring to Figure 5, for example, it shows an OED ~~504~~ 500 describing the main FLIGHTS window 201a shown in Figure 3 of the web page 200 of Figure 2. As noted above, this exact window may be used in a number of the pages of the web site. The main flights window OED ~~500a~~ 500 is developed by first drawing a window object symbol 501. Next, a big circle 503 is drawn around window symbol 501 to define it as the main object of OED 500. Preferably, inheritance is represented within the big circle 503. Accordingly, inheritance symbol 505 and the class that is the source of the inherited characteristics is drawn within the circle 503. In this case, the source is the class "windows", and it is represented by a class symbol 507. The inheritance symbol 505 should be drawn between the object inheriting the features, namely, window object 501, and the object from which it is inheriting those features, namely, class object 507, with the arrow pointing toward the object that is inheriting the features. Event scripts that are executed upon the opening or closing of the main object, i.e., the FLIGHTS window 501, also may be placed within the big circle 503. In this particular example, there is a script for each of those events and they are represented in the drawing by script symbols 509 and 511 corresponding to scripts executed upon opening and closing, respectively, of the FLIGHTS window 501. The SCRIPTS 509 and 511 are connected by a simple line to the object to which they are assigned. Event script symbols such as symbols 509 and 511 in Figure 5 do not represent the code module, e.g., method, that is invoked by the script. They represent the script. The method invoked by the event script is

separately represented in an OED with a method type symbol. The OED or collection of OEDs that represent an application should show that the method invoked by the event script is available to the main object of the OED on which the event script appears. Thus, as will be seen from the discussion further below, the relevant method should be shown either in this OED linked to window symbol 501 or, alternatively, in the OED of another object from which the main object, window 501, of this OED has inheritance.

Page 16, amend Paragraph 0063 as follows:

In addition to the buttons discussed above, the FLIGHTS window 201a has a menu bar 221. A mouse click on each menu will invoke some other program module. Menu symbol 521 represents the menu bar 221 in the FLIGHTS window. If the architect thinks that the menus are sufficiently complex and/or non-standard, he or she may provide a separate, additional OED for the menu bar. Alternately, the application architect simply may provide a textual description of the menus that comprise the menu bar (which could be provided in a separate document or embedded within the OED by any of the previously described techniques). As an even further alternative, the architect may have, instead, decided to show each menu in the menu bar as a separate menu symbol directly within the OED ~~300a~~ 500, rather than as a single menu symbol representing the entire menu bar as shown. Any of these options (as well as many other options) for illustrating the desired logic would be sufficient to describe the logic to a programmer.

Page 17, amend Paragraph 0067 as follows:

Again, the functionality of the actual methods represented by the method symbols should be described in a separate document or by text hidden or otherwise embedded within the OED. However, by way of exemplification, for instance, the VALIDATE AIRLINE method might be a code module that checks the name of an airline entered by a user of the web page 200 to assure that the name corresponds to a known airline and, if it does not, inform the user of that fact and ask the user to enter a proper airline. As another example, the GET DATABASE method may connect to a database. A database connector is needed to build a list of potential flight plans for a user. The HANDLE PRINTING method may include, for instance, functions such as formatting the data, sending the data to a printer, monitoring the printer status, etc. The HANDLE PRINTING method object ~~389~~ 555 is likely to be a complex object in the sense that it likely has many other objects assigned to or defined within it. Accordingly, an application architect may decide to prepare another OED to set forth the details of the HANDLE PRINTING object. If so, a reference to the other OED might be placed within the HANDLE PRINTING method symbol 555 in any of the manners previously mentioned.

Page 18, amend Paragraph 0069 as follows:

Turning now to Figure 6, it is the OED 600 for FLIGHTS RESULTS window 201b. Accordingly, it comprises an appropriately labeled window symbol

601 in the center of a large circle 602, indicating that the FLIGHTS RESULTS window 601 is the main object of this particular diagram 600. The FLIGHTS RESULTS window inherits the properties of the FLIGHTS window ~~304~~ 501, as demonstrated by drawing the symbol 501 for the FLIGHTS window within the circle 602 and connecting it to the flights results window with an inheritance symbol 603.

Page 20, amend Paragraph 0073 as follows:

Frame 290 in Figures 2 and 4 is represented by frame symbol 690 in the OED ~~300b~~ 600 of Figure 6. This is the frame in which flight plan records that are built responsive to the query parameters entered by the user in the aforementioned text boxes in window 201b will be displayed. Frame object 690 is a somewhat complex object and, therefore, has its own OED, which is shown in Figure 7 and will be discussed in detail further below. However, this OED shows some details about frame object 690. Specifically, it shows that an event, namely, a double mouse click (termed “dclick” in the diagrams) over a particular flight plan record 291 listed in the frame will invoke an event script. This is represented in the OED by event script symbol 611. Event script 611 will invoke a method called START RESERVATIONS. THE START RESERVATIONS method is not represented by the event script symbol 611. Nevertheless, by way of explanation, that method would, for example, open a RESERVATIONS window (shown in Figure 7, which is discussed in detail below) within which the user can book the selected flight.